Supplementary Material for Sliced Optimal Partial Transport

1 Relation Between Optimal Partial Transport and Unbalanced Optimal Transport

If we replace the penalty term of (3) with a constraint, i.e. we impose the condition $\gamma(\Omega^2) \geq M$, and use the fact that $|\pi_{1\#}\gamma| = |\pi_{2\#}\gamma| = \gamma(\Omega^2)$, then (3) is closely related to the Lagrangian formulation of the following "primal problem":

$$\text{Primal-OPT}(\mu, \nu; M) = \inf_{\gamma \in \mathcal{M}_+(\Omega^2)} \int c(x, y)\, \mathrm{d}\gamma(x, y) \quad \text{s.t. } \gamma(\Omega^2) \geq M. \tag{7}$$

This, in turn, is closely related to the optimal partial transport problem proposed by [4, 11, 10] (the difference being the mass constraint of γ is imposed as an equality $\gamma(\Omega^2) = M$ rather than a lower bound $\gamma(\Omega^2) \geq M$).

Another equivalent form of the OPT problem defined in (3) is the "generalized Wasserstein distance" in [16, 17] (We refer to [5, Proposition 1.1] and [16, Proposition 4] for their equivalence.) Recently, more systematic studies of so-called "unbalanced optimal transport" or "optimal entropy transportation" problems have been conducted, for instance, in [7] and [15]. OPT, (3), can be seen as a special case of these models, see for instance [7, Theorem 5.2]. It is also well known that in addition to the static Kantorovich formulations presented here, one can also give equivalent dynamic formulations in the spirit of the Benamou–Brenier formula, e.g. [7]. Finally, a related class of models with close relations to the POT problem is discussed in [14] under the name "Generalized Unnormalized Optimal Transport"(GUOT).

2 Relation Between Optimal Partial Transport and Optimal Transport

Inspired by Caffarelli et al.s' technique [4], suppose $\Omega = \mathbb{R}^d$, we introduce an *isolated point* $\hat{\infty}$ into Ω by letting $\hat{\Omega} = \Omega \cup \{\hat{\infty}\}$. Suppose $\hat{\mu} = \mu + (K - \mu(\Omega))\delta_{\hat{\infty}}$, where $\delta_{\hat{x}}$ is the Dirac mass at $\hat{x} \in \hat{\Omega}$ and $\hat{\nu} = \nu + (K - \nu(\Omega))\delta_{\hat{\infty}}$, where the constant K satisfies $K \geq \mu(\Omega) + \nu(\Omega)$, and $\hat{c}(x, y) : \hat{\Omega} \times \hat{\Omega} \to \mathbb{R}_+$ is defined as

$$\hat{c}(x, y) := \begin{cases} c(x, y) - 2\lambda & \text{if } x, y \neq \hat{\infty} \\ 0 & \text{otherwise.} \end{cases}$$

We introduce the following optimal transport problem:

$$\inf_{\hat{\gamma} \in \Gamma(\hat{\mu}, \hat{\nu})} \int \hat{c}(x, y)\, \mathrm{d}\hat{\gamma}(x, y) \tag{8}$$

We claim there exists an equivalence between this OT problem and $\text{OPT}_\lambda(\mu, \nu)$.

Proposition 2.1. *The mapping:* $T : \Gamma_\leq(\mu, \nu) \to \Gamma(\hat{\mu}, \hat{\nu})$ *defined by*

$$\gamma \mapsto \hat{\gamma} = \gamma + (\mu - (\pi_1)_\#\gamma) \otimes \delta_{\hat{\infty}} + \delta_{\hat{\infty}} \otimes (\nu - (\pi_2)_\#\gamma) + (\gamma(\Omega^2) + \alpha)\delta_{\hat{\infty}, \hat{\infty}}, \tag{9}$$

is a bijection, where $\alpha = K - (\mu(\Omega) + \nu(\Omega))$ *and* γ *is optimal in* $\text{OPT}_\lambda(\mu, \nu)$ *if and only if* $\hat{\gamma}$ *is optimal in* (8).

Proof. First we will show that $\hat{\gamma} = T(\gamma) \in \Gamma(\hat{\mu}, \hat{\nu})$ for $\gamma \in \Gamma_{\leq}(\mu, \nu)$. Pick a Borel set $A \subset \hat{\Omega}$, and suppose $\hat{\infty} \in A$. By definition, $\hat{\gamma}$ is a measure defined on $\hat{\Omega}^2$, then we have

$$\begin{aligned}
\hat{\gamma}(A \times \hat{\Omega}) &= \hat{\gamma}(A \setminus \{\hat{\infty}\} \times \Omega) + \hat{\gamma}(A \setminus \{\hat{\infty}\} \times \{\hat{\infty}\}) + \hat{\gamma}(\{\hat{\infty}\} \times \Omega) + \hat{\gamma}(\{\hat{\infty}, \hat{\infty}\}) \\
&= \gamma(A \setminus \{\hat{\infty}\} \times \Omega) + (\mu - (\pi_1)_{\#}\gamma)(A \setminus \{\hat{\infty}\}) + (\nu - (\pi_2)_{\#}\gamma)(\Omega) + \gamma(\Omega^2) + \alpha \\
&= (\pi_1)_{\#}\gamma(A \setminus \{\hat{\infty}\}) + (\mu - (\pi_1)_{\#}\gamma)(A \setminus \{\hat{\infty}\}) + \nu(\Omega) - (\pi_2)_{\#}\gamma(\Omega) + \gamma(\Omega^2) + \alpha \\
&= \mu(A \setminus \{\hat{\infty}\}) + \nu(\Omega) + \alpha \\
&= \hat{\mu}(A)
\end{aligned}$$

Similarly, if $\hat{\infty} \notin A$, we have $\hat{\gamma}(A \times \hat{\Omega}) = \mu(A) = \hat{\mu}(A)$. Thus $(\pi_1)_{\#}\hat{\gamma} = \hat{\mu}$ and similarly $(\pi_2)_{\#}\hat{\gamma} = \hat{\nu}$. Therefore $\hat{\gamma} \in \Gamma(\hat{\mu}, \hat{\nu})$.

It is obvious that the mapping T is injective since if $\gamma_1 \neq \gamma_2$ where $\gamma_1, \gamma_2 \in \Gamma_{\leq}(\mu, \nu)$, then there exists one set $B \subset \Omega^2$ such that $\gamma_1(B) \neq \gamma_2(B)$. Then $\hat{\gamma}_1(B) = \gamma_1(B) \neq \gamma_2(B) = \hat{\gamma}_2(B)$. Therefore, $\hat{\gamma}_1 \neq \hat{\gamma}_2$.

Next, we will show the surjectivity of T. Pick any $\hat{\gamma} \in \Gamma(\hat{\mu}, \hat{\nu})$, define γ such that for any $B \subset \Omega^2$, $\gamma(B) = \hat{\gamma}(B)$. We have $\gamma \in \Gamma_{\leq}(\mu, \nu)$. Indeed, pick Borel set $A \subset \Omega$, we have

$$\gamma(A \times \Omega) = \hat{\gamma}(A \times \Omega) \leq \hat{\gamma}(A \times \hat{\Omega}) = \hat{\mu}(A) = \mu(A).$$

Thus $(\pi_1)_{\#}\gamma \leq \mu$, similarly we have $(\pi_2)_{\#}\gamma \leq \nu$. Let $\hat{\gamma}_1 = T(\gamma)$. We claim $\hat{\gamma} = \hat{\gamma}_1$.Note, since $\Omega^2, \Omega \times \{\hat{\infty}\}, \{\hat{\infty}\} \times \Omega, \{\hat{\infty}, \hat{\infty}\}$ is a disjoint decomposition of $\hat{\Omega}^2$ (and all of them are measurable), it is sufficient to prove $\hat{\gamma}(B) = \hat{\gamma}_1(B)$ for any Borel set B which is a subset of one of these four sets.

Case 1: If $B \subset \Omega^2$, we have $\hat{\gamma}_1(B) = \gamma(B) = \hat{\gamma}(B)$.

Case 2: If $B = A \times \{\hat{\infty}\}$ where $A \subset \Omega$ is Borel set, then

$$\begin{aligned}
\hat{\gamma}(B) &= \hat{\gamma}(A \times \hat{\Omega}) - \hat{\gamma}(A \times \Omega) \\
&= \hat{\mu}(A) - \gamma(A \times \Omega) \\
&= \mu(A) - (\pi_1)_{\#}\gamma(A) \\
&= \hat{\gamma}_1(A \times \{\hat{\infty}\}) = \hat{\gamma}_1(B)
\end{aligned}$$

Similarly, if $B = \{\hat{\infty}\} \times A$ for some $A \subset \Omega$, we have $\hat{\gamma}(B) = \hat{\gamma}_1(B)$.

Case 3: If $B = \{(\hat{\infty}, \hat{\infty})\}$. Note, since $\hat{\gamma}_1 \in \Gamma(\hat{\mu}, \tilde{\nu})$ as we discussed above, then $\hat{\gamma}(\hat{\Omega}^2) = \hat{\gamma}_1(\hat{\Omega}^2)$. Additionally, by Cases 1 and 2 we have $\hat{\gamma}(\Omega \times \Omega) = \hat{\gamma}_1(\Omega \times \Omega)$, $\hat{\gamma}(\Omega \times \{\hat{\infty}\}) = \hat{\gamma}_1(\Omega \times \{\hat{\infty}\})$ and $\hat{\gamma}(\{\hat{\infty}\} \times \Omega) = \hat{\gamma}_1(\{\hat{\infty}\} \times \Omega)$. Thus

$$\begin{aligned}
\hat{\gamma}(B) &= \hat{\gamma}(\hat{\Omega}^2) - \hat{\gamma}(\Omega^2) - \hat{\gamma}(\Omega \times \{\hat{\infty}\}) - \hat{\gamma}(\{\hat{\infty}\} \times \Omega) \\
&= \hat{\gamma}_1(\hat{\Omega}^2) - \hat{\gamma}_1(\Omega^2) - \hat{\gamma}_1(\Omega \times \{\hat{\infty}\}) - \hat{\gamma}_1(\{\hat{\infty}\} \times \Omega) \\
&= \hat{\gamma}_1(B)
\end{aligned} \tag{10}$$

Hence, $\hat{\gamma} = \hat{\gamma}_1$ and thus that the mapping is surjective.

We will show γ is optimal in $\mathrm{OPT}_\lambda(\mu, \nu)$ if and only if $\hat{\gamma}$ is optimal in $\mathrm{OT}(\hat{\mu}, \hat{\nu})$ (defined in (8)). We let $C(\gamma), \hat{C}(\hat{\gamma})$ denote the corresponding transportation cost of $\gamma, \hat{\gamma}$ with respect to the OPT, OT problems, i.e.

$$\hat{C}(\hat{\gamma}) = \int \hat{c}(x, y)\, \mathrm{d}\gamma(x, y), \qquad C(\gamma) = \int c(x, y)\, \mathrm{d}\gamma(x, y) + \lambda \left(\mu(\Omega) - \pi_{1\#}\gamma(\Omega) + \nu(\Omega) - \pi_{2\#}\gamma(\Omega)\right). \tag{11}$$

We have $C(\gamma) = C(\hat{\gamma}) + \lambda(\mu(\Omega) + \nu(\Omega))$. Combined with the fact the mapping is a bijection, we have γ is optimal iff $\hat{\gamma}$ is optimal. □

3 Proofs in Section 3

Proof of Proposition 3.1. Let $\hat{\gamma}$ be optimal for the extended balanced problem of the previous section, (8), and let γ be the restriction of this measure to $\Omega \times \Omega$. Since restriction preserves optimality [21, Theorem 4.6], γ must be an optimal plan between $\pi_{1\#}\gamma$ and $\pi_{2\#}\gamma$ with respect to the (non-extended) cost c on $\Omega \times \Omega$.

Therefore, it must be supported on a c-cyclically monotone set [21, Theorem 5.10]. In one dimension, for costs of the form $c(x,y) = f(x-y)$ for convex f, c-cyclical monotonicity reduces to standard monotonicity, see for instance [19, Theorem 2.9]. □

Proof of Lemma 3.2. Pick γ and define γ' as follows: for any Borel $A \subset \Omega^2$, $\gamma'(A) = \gamma(A \setminus S)$. Let

$$\begin{aligned} C(\gamma) &:= \int c(x,y)\,\mathrm{d}\gamma + \lambda\left[(\mu(\Omega) - \pi_{1\#}\gamma(\Omega)) + (\nu(\Omega) - \pi_{2\#}\gamma(\Omega))\right] \\ &= \int (c(x,y) - 2\lambda)\ \mathrm{d}\gamma + \lambda(\mu(\Omega) + \nu(\Omega)), \end{aligned}$$

which is the objective function for the OPT problem defined in (3), and the second line follows from the fact $\gamma(\Omega^2) = (\pi_1)_\#\gamma(\Omega) = (\pi_2)_\#\gamma(\Omega)$. Then we have

$$C(\gamma) - C(\gamma') = \int_S (c(x,y) - 2\lambda)\,\mathrm{d}\gamma(x,y) \geq 0$$

That is, for any γ, we can find a better transportation plan γ' such that $\gamma'(S) = 0$. □

4 Proofs in Section 4

Proof of Theorem 4.1. We start by adapting the extension (8) to the concrete discrete setting between empirical measures. Let $\hat{\Omega} = [1 : m+n]$, and let $\hat{c} : \hat{\Omega} \times \hat{\Omega}$ be given by

$$\hat{c}(i,j) = \begin{cases} c(x_i, y_j) - 2\lambda & \text{if } i \leq n, j \leq m, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\hat{\mu} = \hat{\nu} = \sum_{i=1}^{m+n} \delta_i$. Then solving the (balanced) optimal transport problem on $\hat{\Omega}$ between $\hat{\mu}$ and $\hat{\nu}$ with respect to cost $\hat{c}$ is (as above) clearly equivalent to the OPT problem, and an optimal OPT plan can be obtained by restricting an optimal $\hat{\gamma}$ to the set $[1:n] \times [1:m]$. Note that here we have simply split the mass on the isolated point $\hat{\infty}$ onto multiple points, where each only carries unit mass. At the same time, the set $\Gamma(\hat{\mu}, \hat{\nu})$ are the doubly stochastic matrices and by the Birkhoff-von-Neumann theorem its extremal points are permutation matrices. Thus there always exists an optimal $\hat{\gamma}$ that is a permutation matrix, and thus its restriction to $[1:n] \times [1:m]$ will only contain entries 0 or 1, with at most one 1 per row and column. □

Proof of Proposition 4.3. For $\lambda_1 = \lambda_2$ we can write (2) as

$$\mathrm{OPT}_\lambda(\mu,\nu) = \mathrm{ET}(\mu,\nu;\mathcal{F},\mathcal{F}) = \inf_{\gamma \geq 0} \int_{\Omega^2} c\mathrm{d}\gamma + \mathcal{F}(\pi_{1\#}\gamma \parallel \mu) + \mathcal{F}(\pi_{2\#}\gamma \parallel \nu)$$

where

$$\mathcal{F}(\hat{\mu} \parallel \mu) = \begin{cases} \lambda \cdot (\mu(\Omega) - \hat{\mu}(\Omega)) & \text{if } 0 \leq \hat{\mu} \leq \mu, \\ +\infty & \text{otherwise.} \end{cases}$$

In particular, $\mathcal{F}$ is the f-divergence associated with integrand

$$F(s) = \begin{cases} \lambda(1-s) & \text{if } s \in [0,1] \\ +\infty & \text{else.} \end{cases}$$

By [15, Theorem 4.11] the dual of ET is

$$\sup_{\substack{\Phi \in \mathrm{L}^1(\mu), \Psi \in \mathrm{L}^1(\nu) \\ \Phi \oplus \Psi \leq c \\ \Phi, \Psi \text{ lsc and bounded}}} -\int_\Omega F^*(-\Phi)\,\mathrm{d}\mu - \int_\Omega F^*(-\Psi)\,\mathrm{d}\nu$$

where

$$F^*(r) = \sup_s (rs - F(s)) = \max\{-\lambda, r\}.$$

By [15, Theorem 4.6] the optimality conditions are

$$\begin{aligned} &\Phi \oplus \Psi = c && && \gamma\text{-a.e.} \\ &-\Phi \in \partial F\left(\frac{\mathrm{d}\gamma_1}{\mathrm{d}\mu}\right), \quad \gamma_1 = \pi_{1\#}\gamma && && \mu\text{-a.e.} \\ &-\Psi \in \partial F\left(\frac{\mathrm{d}\gamma_2}{\mathrm{d}\nu}\right), \quad \gamma_2 = \pi_{2\#}\gamma && && \nu\text{-a.e.} \end{aligned} \tag{12}$$

We have

$$\partial F(s) = \begin{cases} \{-\lambda\} & \text{if } s \in (0,1) \\ (-\infty, -\lambda] & \text{if } s = 0 \\ [-\lambda, +\infty) & \text{if } s = 1. \end{cases}$$

So (12) can be written

$$\begin{aligned} &\Phi(x) = \lambda && \text{if } \frac{\mathrm{d}\gamma_1}{\mathrm{d}\mu}(x) \in (0,1) \\ &\Phi(x) \in [\lambda, +\infty) && \text{if } \frac{\mathrm{d}\gamma_1}{\mathrm{d}\mu}(x) = 0 \\ &\Phi(x) \in (-\infty, \lambda] && \text{if } \frac{\mathrm{d}\gamma_1}{\mathrm{d}\mu}(x) = 1. \end{aligned}$$

Similarly for Ψ. In the discrete case the dual problem and optimality conditions reduce to the form stated in the proposition. □

5 Correctness and complexity of Algorithms 1 and 2

5.1 Correctness

In this section, we prove the correctness of Algorithms 1 and 2 as stated above, and we discuss how to deal with duplicate points. Extended versions of the Algorithms with more sophisticated data structures and proper handling of boundaries and duplicates are then given in Section 5.2 together with a bound on their worst-case complexity.

Preliminaries, induction strategy, cases 1 and 2. Throughout this proof, we are simply going to write $c_{i,j}$ for $c(x_i, y_j)$. We assume that the point lists $\{x_i\}_{i=1}^n$ and $\{y_j\}_{j=1}^m$ are sorted, but we now allow for duplicate points and their handling will be addressed throughout this proof. Since $c(x,y) = h(x-y)$ for h strictly convex, it is easy to verify that

$$c_{i,j} + c_{k,l} \leq c_{i,l} + c_{k,j} \tag{13}$$

if $i \leq k$ and $j \leq l$, with a strict inequality if $x_i < x_k$ and $y_j < y_l$. This is known as Monge property [3]. The proof works via induction in the iterations of the main loop of Algorithm 1. We will show that prior to the iteration for x_k / after completing the iteration for x_{k-1}, the following holds:

I. $\Psi_j \leq \lambda$ for all $j \in [1:m]$.

II. For all $j \in [1:m]$, if $\Psi_j < \lambda$, then y_j is assigned.

III. $\Phi_i \leq \lambda$ for all $i \in [1:n]$.

IV. For all $i \in [1:k-1]$, if $\Phi_i < \lambda$, then x_i is currently assigned.

V. All dual constraints $\Phi_i + \Psi_j \leq c_{i,j}$ for all $i \in [1:n]$, $j \in [1:m]$ hold.

VI. For all $i \in [1:n]$, $j \in [1:m]$, whenever x_i is assigned to y_j, one has $\Phi_i + \Psi_j = c_{i,j}$.

VII. The assignment L will be monotonous. I.e. if $L[i] \neq -1$, $L[i'] \neq -1$ for $i < i'$, then $L[i] < L[i']$.

We initialize with $\Psi_j = \lambda$ for all j, $\Phi_i = -\infty$ for all i, and empty assignment $L_i = -1$ for all i. Therefore, prior to the first iteration, when $k = 0$, all conditions are satisfied. Next, note that items (I) and (II) will be satisfied throughout the algorithm since entries of Ψ are only ever decreased during the algorithm; entries are only decreased when the corresponding y_j are assigned; and once a point y_j is assigned, it may become re-assigned, but it never becomes un-assigned again. The claim that y_j is never un-assigned is clear in all cases apart from Case 3.1. In Case 3.1 it follows from property (VIII) below, which implies that when x_{i_Δ} is un-assigned from, say, y_{j_Δ}, then y_{j_Δ} is re-assigned to $i_\Delta + 1$ (since the assignment between $x_{i'}$ and $y_{j'}$ satisfies $L[i'] = j_{\min} + i' - i_{\min}$, i.e. the assignment is consecutive).

Throughout the algorithm, let j_{last} be the largest index among any assigned points y_j. We initially set $j_{\text{last}} = -1$ when no y_j is assigned. Since assigned y_j does not get un-assigned (merely re-assigned), j_{last} is non-decreasing.

Lemma 5.1. *If $j_{\text{last}} \neq -1$, then during the iteration of the main loop of Algorithm 1, for any minimizer j^* in line 3, one has $y_{j^*} \geq y_{j_{\text{last}}}$. In particular, j^* can always be chosen such that $j^* \geq j_{\text{last}}$.*

Proof. If $j_{\text{last}} = -1$, there is nothing to prove, since $j^* \geq 1$. If $j_{\text{last}} \neq -1$, then there must be some $i \in [1 : k-1]$ such that $L[i] = j_{\text{last}}$ and therefore $\Phi_i + \Psi_{j_{\text{last}}} = c_{i,j_{\text{last}}}$. After adjusting Φ_k in line 4 one has $\Phi_k + \Psi_{j^*} = c_{k,j^*}$. By dual feasibility, we have in addition

$$\Phi_k + \Psi_{j_{\text{last}}} \leq c_{k,j_{\text{last}}}, \qquad \Phi_i + \Psi_{j^*} \leq c_{i,j^*}.$$

Combining these four (in-)equalities we get

$$c_{k,j^*} + c_{i,j_{\text{last}}} \leq c_{k,j_{\text{last}}} + c_{i,j^*}.$$

If $x_k > x_i$, then by (13) we have $y_{j^*} \geq y_{j_{\text{last}}}$. So $j^* < j_{\text{last}}$ can only happen if $y_{j^*} = y_{j_{\text{last}}}$ and thus we may also choose j_{last} as minimizing index. Therefore, we may impose the constraint $j^* \geq j_{\text{last}}$ in line 3. In the case $x_k = x_i$, assume j_{last} would not be a minimal index in line 3, i.e.

$$c_{k,j^*} - \Psi_{j^*} < c_{k,j_{\text{last}}} - \Psi_{j_{\text{last}}} = \Phi_i,$$

where in the last equality, we used $x_k = x_i$. Since $L[i] = j_{\text{last}}$, one must have that the dual constraint for (i, j_{last}) must be active. This would imply that the dual constraint for (i, j^*) is violated, which contradicts the induction hypothesis. □

Now during iteration k, the change of Φ_k in line 4, by construction, preserves (III) and (V). Assume we enter Case 1. The assignment function L is not changed, hence (VI) and (VII) remain preserved, and since $\Phi_k = \lambda$, (IV) is extended to $i = k$. Assume we enter Case 2. Then we have $L_k = j^*$, $\Phi_k + \Psi_{j^*} = c_{k,j^*}$ and $\Phi_k < \lambda$. Hence, (VI) remains true, and (IV) is extended to $i = k$. If we choose $j^* > j_{\text{last}}$ (which is possible by Lemma 5.1), we preserve (VII).

Case 3. We now turn to Case 3.

Lemma 5.2. *In each iteration of the main loop of Algorithm 1, when we enter Case 3, i.e. $\Phi_k < \lambda$ and $j^* = j_{\text{last}}$, let i be the index such that x_i is currently assigned to $y_{j_{\text{last}}}$. Then $x_i = x_{i'}$ for all $i' \in [i : k-1]$. If $i < k-1$, then one must have $\Phi_i = \lambda$.*

Proof. Clearly, $i < k$ (since it was assigned during a previous iteration). In the following, let $f(x) = c(x, y_{j_{\text{last}}}) - \Psi_{j_{\text{last}}}$, which is convex in $x \in \mathbb{R}$. By (VI) we have $\Phi_i = c(x_i, y_{j_{\text{last}}}) - \Psi_{j_{\text{last}}} = f(x_i) \leq \lambda$, by the current iteration of the main loop we have $\Phi_k + \Psi_{j_{\text{last}}} = c(x_k, y_{j_{\text{last}}}) = f(x_k) < \lambda$. Let now $i' \in [i+1, k-1]$. By monotonicity of L, (VII), if $L[i'] \neq -1$, then we would need $L[i'] > j_{\text{last}}$, which contradicts the definition of j_{last}. Therefore $L[i'] = -1$ and therefore by (IV) we must have $\Phi_{i'} = \lambda$. By (V) we must also have $\Phi_{i'} \leq f(x_{i'})$, and by convexity of f, $f(x_{i'})$, since $f(x_i) \leq \lambda$, $f(x_k) < \lambda$, this can only happen if $x_{i'} = x_i$, and $\Phi_i = f(x_i) = \lambda$. □

This means that if all points are distinct, then we must have $i = k-1$, and find ourselves in the main loop of Algorithm 2, see below.

Remark 5.3. *If points are not necessarily distinct (at least up to numerical rounding errors), and we find $i < k-1$, then the situation can be remedied by setting $L[i] = -1$, $L[k] = j^*$, which preserves (VI), (IV) and (VII). We will add this to the algorithms in Section 5.2.*

We now study Algorithm 2 to resolve the conflict. In addition to the items above, at the beginning of each iteration of the main loop of Algorithm 2 the following is preserved:

VIII. There are indices $j_{\min} \le j^*$, $i_{\min} \le k-1$ with $j^* - j_{\min} = (k-1) - i_{\min}$ such that $L[i_{\min}+r] = j_{\min}+r$ for $r \in [0 : (k-1) - i_{\min}]$, $\Phi_{i+r} + \Psi_{j_{\min}+d-1} = c_{i+r,j_{\min}+d-1}$ for $r \in [1 : (k-1) - i_{\min}]$.

This is clearly true before the first iteration, when $i_{\min} = k-1$ and $j_{\min} = j^*$. In each iteration of the main loop we then seek the largest possible value $\Delta \ge 0$ such that by setting $\Phi_i \leftarrow \Phi_i + \Delta$ for $i \in [i_{\min}, k]$ and $\Psi_j \leftarrow \Psi_j - \Delta$ for $j \in [j_{\min}, j^*]$ we preserve all items (I) to (VII). Clearly the delicate ones are (III) and (V). To preserve the former, we ensure that $\Delta \le \lambda_\Delta$. To preserve the latter, we do not need to worry about the Ψ_j, $j \in [j_{\min} : j^*]$, since they are decreased, but we need to consider all constraints $\Phi_i + \Psi_j \le c_{i,j}$ for $i \in [i_{\min} : k]$, $j \in [1 : j_{\min} - 1] \cup [j^* + 1 : m]$. By the following lemma, we see that this can be reduced to checking the two constraints for $(i_{\min}, j_{\min} - 1)$ and $(k, j^* + 1)$, which is the role of the variables α and β in Algorithm 2.

Lemma 5.4. *In the above situation, one has that*

$$\min_{\substack{i\in[i_{\min}:k],\\ j\in[j^*+1:m]}} c_{i,j} - \Phi_i - \Psi_j = c_{k,j^*+1} - \Phi_k - \Psi_{j^*+1},$$

$$\min_{\substack{i\in[i_{\min}:k],\\ j\in[1:j_{\min}-1]}} c_{i,j} - \Phi_i - \Psi_j = c_{i_{\min},j_{\min}-1} - \Phi_{i_{\min}} - \Psi_{j_{\min}-1},$$

if $j^* < m$ and $j_{\min} > 1$ respectively.

Proof. We start with the first equation and begin by showing that

$$c_{i,j} - \Phi_i - \Psi_j \ge c_{k,j} - \Phi_k - \Psi_j \tag{14}$$

for $i \in [i_{\min} : k]$, $j \in [j^* + 1 : m]$. We get this by combining $\Phi_i \le c_{i,j^*} - \Psi_{j^*}$, $\Psi_{j^*} = c_{k,j^*} - \Phi_k$ and the Monge property of cost matrix, (13), $c_{k,j} \le c_{i,j} + c_{k,j^*} - c_{i,j^*}$. Next, observe that $\Psi_j = \lambda$ for $j \in [j^* + 1 : m]$, since these values have not yet been changed since the initialization, and $\Psi_{j^*} \le \lambda$. Also we know that $\Phi_k = c_{k,j^*} - \psi_{j^*} \le c_{k,j} - \psi_j$ for all $j \in [1 : m]$. Combining this, we get $c_{k,j^*} \le c_{k,j}$ for $j \in [j^* + 1 : m]$. Since $f : y \mapsto c(x_k, y)$ is convex, and $f(y_{j^*}) \le f(y_j)$ for $j \ge j^*$, we must have that f is non-decreasing after y_{j^*}, and therefore among all indices $j \ge j^*$, the smallest one attains the minimum.

Now we turn to the second equation. In complete analogy to (14) we show that $c_{i,j} - \Phi_i - \Psi_j \ge c_{i_{\min},j} - \Phi_{i_{\min}} - \Psi_j$ for $i \in [i_{\min} : k]$, $j \in [1 : j_{\min} - 1]$. Arguing as in Lemma 5.1, we can show a minimizing index j can be chosen such that it is not smaller than $\hat{j}$, where $\hat{j}$ is the largest index among the assigned y_j, that is less than $j_{\min}$ (if such an assigned point exists, otherwise just let $\hat{j} = 0$ in the following). Consequently, all y_j points in $[\hat{j}+1 : j_{\min}-1]$ must be unassigned and therefore have $\Psi_j = \lambda$. Arguing then via the convexity of c as in the previous paragraph, we can show that a minimizing j must be given by $j_{\min} - 1$. □

The selection of Cases 3.1, 3.2 or 3.3 depends now on which of the three bounds λ_Δ, α, or β is smallest. Consequently, each of the implied updates of the dual variables in the three cases preserves the dual constraints and it is easy to see that by property (VIII) each of the conflict resolutions in Cases 3.1, 3.2 and 3.3a preserve all other conditions (I) to (VII). For instance, when λ_Δ is minimal, element x_{i_Δ} becomes unassigned, however we then have $\Phi_{i_\Delta} = \lambda$ as required by (IV).

We are left with discussing Case 3.3b, i.e. when β is minimal among the three bounds and $y_{j_{\min}-1}$ is already assigned. In complete analogy to Lemma 5.2 we can prove the following.

Lemma 5.5. *In each iteration of the main loop of Algorithm 2, when we enter Case 3.3b, i.e. $\Phi_{i_{\min}} < \lambda$ and $y_{j_{\min}-1}$ is already assigned, let i be the index such that x_i is currently assigned to $y_{j_{\min}-1}$. Then $x_i = x_{i'}$ for all $i' \in [i : i_{\min} - 1]$. If $i < i_{\min} - 1$, then one must have $\Phi_i = \lambda$.*

As above, this means that if all points are distinct, then $i = i_{\min} - 1$, we can set $i_{\min} \leftarrow i_{\min} - 1$, $j_{\min} \leftarrow j_{\min} - 1$, note that we satisfy $\Phi_{i_{\min}} + \Psi_{j_{\min}} = c_{i_{\min}, j_{\min}}$ and thus preserve (VIII) before the next iteration in Algorithm 2.

Remark 5.6. *If points are not all distinct and if $i < i_{\min} - 1$, then we must have $\Phi_i = \lambda$, thus we can unassign x_i and $y_{j_{\min}-1}$, and then proceed as if $y_{j_{\min}-1}$ were unassigned and resolve the conflict as in Case 3.3a.*

5.2 Full algorithm versions and complexity

We now give more complete pseudo code versions of the Algorithms 1 and 2, see Algorithms 1 and 2. The main purpose is to reach a quadratic worst case time complexity. Our algorithm can be seen as a specialization of the Hungarian method, exploiting the particular one-dimensional structure of the cost and dealing consistently with the option to discard mass for a cost λ. The changes are explained below, subsequently some additional changes (for duplicate and boundary handling) are described in plain text, and finally we show how to determine the time complexity bound.

Algorithm 1: opt-1d

Input: $\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m, \lambda$
Output: L, Ψ, Φ

```
1  Initialize Φ_i ← −∞ for i ∈ [1 : n], Ψ_j ← λ for j ∈ [1 : m] and L_i ← −1 for i ∈ [1 : n],
2  j_last ← 1
3  for k = 1, 2, ... n do
4      j* ← argmin_{j∈[j_last:m]} c(x_k, y_j) − Ψ_j
5      Φ_k ← min{c(x_k, y_{j*}) − Ψ_{j*}, λ}
6      if Φ_k = λ then
7          [Case 1] No update on L
8      else if j_min − 1 unassigned then
9          [Case 2] L_k ← j*, j_last ← j*
10     else
11         [Case 3] Run Algorithm 2.
```

Implemented modifications compared to Algorithms 1 and 2. Compared to Algorithm 1, in Algorithm 1 we have added the variable j_{last} for improved handling of duplicate points (or limited numerical precision), see Remark 5.3. Note that initializing $j_{\text{last}} \leftarrow 1$, even when no points are yet assigned, yields the desired behaviour. Additional adaptations related to this are discussed in the paragraph below.

Compared to Algorithm 2, there are several adaptations to Algorithm 2.

The dual variables Φ and Ψ are not updated during every loop of the algorithm but only once, when the conflict is resolved. This is handled via the auxiliary variable v and the auxiliary array d. The former stores the total increment that will need to be applied to Φ_k at the end, in addition d_i stores the value of v at the time when i was added to the 'chain', therefore $v - d_i$ will be the necessary increment of Φ_i at the end. This trick (which is also known for the Hungarian method) removes the necessity to loop over the whole chain to update the dual variables during each iteration of the main loop in Algorithm 2 and thus reduces the worst case time complexity from cubic to quadratic.

Similarly, the index of the dual variable Φ_i that is currently closest to λ is not determined from scratch during each iteration. Instead, when case 3.3b is entered, the old best value is first reduced by β, then compared with the new competitor $i_{\min}$ (after updating $i_{\min}$), and updated if necessary.

Additional recommended modifications to algorithm. In lines 5 and 6 of Algorithm 2 boundary checks should be added. E.g. α can only be set as described if $j^* < m$, otherwise it should be set to $+\infty$. Likewise, β can only be set as described if $j_{\min} > 1$ and should otherwise be set to $+\infty$. To keep track of which y_j are assigned one can use a boolean array of size m, initialized with false, and entries corresponding

Algorithm 2: sub-opt-full

Input: $(\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m, k, j^*, j_{\text{last}}, L, \Phi, \Psi)$
Output: (Updated L, Φ, Ψ, optimal for OPT$(\{x_i\}_{i=1}^k, \{y_j\}_{j=1}^m)$, and j_{last})

Initialize $i_{\min} \leftarrow k-1$, $j_{\min} \leftarrow j^*$.
Initialize $v \leftarrow 0$, $d_j \leftarrow 0$ for $j \in [1:m]$, $d_k \leftarrow 0$, $d_{k-1} \leftarrow 0$.
$i_\Delta \leftarrow \text{argmin}_{i \in [k-1:k]}(\lambda - \Phi_i)$, $\lambda_\Delta \leftarrow \lambda - \Phi_{i_\Delta}$
while *true* **do**
  $\alpha \leftarrow c(x_k, y_{j^*+1}) - \Phi_k - v - \Psi_{j^*+1}$
  $\beta \leftarrow c(x_{i_{\min}}, y_{j_{\min}-1}) - \Phi_{i_{\min}} - \Psi_{j_{\min}-1}$
  if $\lambda_\Delta \le \min\{\alpha, \beta\}$ **then**
    [**Case 3.1**]
    $v \leftarrow v + \lambda_\Delta$
    for $i \in [i_{\min}, k-1]$ **do**
      $\Phi_i \leftarrow \Phi_i + v - d_i$, $\Psi_{L_i} \leftarrow \Psi_{L_i} - v + d_i$
    $\Phi_k \leftarrow \Phi_k + v$
    $L_{i_\Delta} \leftarrow -1$, $L_k \leftarrow j^*$
    for $i \in [i_\Delta + 1 : k-1]$ **do**
      $L_i \leftarrow L_i - 1$
    return
  else if $\alpha \le \min\{\lambda_{\text{diff}}, \beta\}$ **then**
    [**Case 3.2**]
    $v \leftarrow v + \alpha$
    for $i \in [i_{\min}, k-1]$ **do**
      $\Phi_i \leftarrow \Phi_i + v - d_i$, $\Psi_{L_i} \leftarrow \Psi_{L_i} - v + d_i$
    $\Phi_k \leftarrow \Phi_k + v$
    $L_k \leftarrow j^* + 1$, $j_{\text{last}} \leftarrow j^* + 1$
    return
  else
    $v \leftarrow v + \beta$
    if $j_{\min} - 1$ *unassigned* **then**
      [**Case 3.3a**]
      for $i \in [i_{\min}, k-1]$ **do**
        $\Phi_i \leftarrow \Phi_i + v - d_i$, $\Psi_{L_i} \leftarrow \Psi_{L_i} - v + d_i$
      $\Phi_k \leftarrow \Phi_k + v$
      $L_{i_{\min}} \leftarrow j_{\min} - 1$, $L_k \leftarrow j^*$
      for $i \in [i_{\min} + 1 : k-1]$ **do**
        $L_i \leftarrow L_i - 1$
      return
    else
      [**Case 3.3b**]
      $d_{i_{\min}-1} \leftarrow v$, $\lambda_\Delta \leftarrow \lambda_\Delta - \beta$,
      $i_{\min} \leftarrow i_{\min} - 1$, $j_{\min} \leftarrow j_{\min} - 1$
      if $\lambda - \Phi_{i_{\min}} < \lambda_\Delta$ **then**
        $\lambda_\Delta \leftarrow \lambda - \Phi_{i_{\min}}$, $i_\Delta \leftarrow i_{\min}$

to assigned points are set to true. This can be used to distinguish between cases 3.3a and 3.3b. Alternatively, an 'inverse' version of L can be maintained, where $L^{-1}[j]$ will store the index i of point x_i to which point y_j is assigned (and -1 otherwise). This has to be updated consistently with L. The latter will be useful when dealing with duplicate points according to Remarks 5.3 and 5.6 at the beginnings of case 3 and case 3.3b respectively.

Worst case time complexity. In Algorithm 1, initialization of the arrays Φ, Ψ and L requires $\Theta(n+m)$ steps. The main loop runs exactly n times. Determining j^* requires $\mathcal{O}(m)$ steps (using the particular structure of c and $\Psi_j = \lambda$ for $j > j^*$ one could reduce this further, see Lemma 5.4, but we leave such optimizations for future work). Cases 1 and 2 take $\Theta(1)$ steps. Let us now consider case 3 and Algorithm 2. Initialization takes $\Theta(m)$ for setting up d (although we note that this initialization could be skipped). Cases 3.1, 3.2 and 3.3a are each entered only once, right before termination of the sub-routine, and they have a complexity of $\mathcal{O}(n)$ (iterating over the chain for a fixed number of times to adjust L and the duals). Case 3.3b, as well as maintaining the variables α, β and λ_Δ have a complexity of $\Theta(1)$ per iteration and there are $\mathcal{O}(n)$ iterations. Hence, Algorithm 2 in its current form has a complexity of $\mathcal{O}(\max\{m,n\})$, and therefore finally Algorithm 1 has a complexity of $\mathcal{O}(n \cdot \max\{m,n\})$. During the proof we have pointed out some potential for optimizing the algorithm for the regime when $n \ll m$.

6 OPT defines a metric

When the cost function $c(x,y)$ is the p-th power of a metric, similar to OT, OPT can also define a metric in $\mathcal{M}_+(\Omega)$. For finite discrete μ, ν, a similar result has been proposed by [13, Theorem 2.2]. Here we propose a more general version:

Theorem 6.1 (OPT defines a metric)**.** *If $c(x,y) : \Omega^2 \to \mathbb{R}_+$ is defines as $c(x,y) = D^p(x,y)$ for some metric D defined on Ω and $\lambda > 0$, then $(OPT_\lambda(\cdot,\cdot))^{1/p}$ defines a metric in $\mathcal{M}_+(\Omega)$.*

Proof. It is straightforward to show $(OPT_\lambda(\cdot,\cdot))^{1/p}$ is symmetric and $(OPT_\lambda(\mu,\nu))^{1/p} = 0$ if and only if $\mu = \nu$. For the triangle inequality, let $\tilde{\Omega}, \tilde{\infty}, \tilde{\mu}, \tilde{\nu}, K$ denote the corresponding concepts as defined in section 2. By Lemma 3.2, we can replace the cost function $D^p(x,y)$ by $D^p(x,y) \wedge 2\lambda$ in the OPT problem, and its optimal value is unchanged. That is

$$\mathrm{OPT}_\lambda(\mu,\nu) = \inf_{\pi \in \Pi_\leq(\mu,\nu)} (D^p(x,y) \wedge 2\lambda) d\gamma + \lambda((\mu(\Omega) - (\pi_1)_\# \gamma(\Omega)) + (\nu(\Omega) - (\pi_2)_\# \gamma(\Omega))).$$

In addition, by proposition 2.1, we have

$$\gamma \mapsto \hat{\gamma} = \gamma + (\mu - \gamma_0) \otimes \delta_{\hat{\infty}} + \delta_{\hat{\infty}} \otimes (\nu - \gamma_1) + (|\gamma| + \alpha)\delta_{\hat{\infty},\hat{\infty}}$$

is a bijection between $\Pi_\leq(\mu,\nu)$ and $\Pi(\hat{\mu},\hat{\nu})$, where $\alpha = K - (|\mu| + |\nu|)$. Let $C(\gamma;\mu,\nu,\lambda)$ denote the objective function of $OPT_\lambda(\mu,\nu)$. Follows the section 3.1 in [13], we define $D'(x,y) : \Omega \cup \{\hat{\infty}\} \to \mathbb{R}$ such that

$$(D')^p(x,y) = \begin{cases} D^p(x,y) \wedge 2\lambda & \text{if } (x,y) \in \Omega \\ \lambda & \text{if } x \in \Omega, y = \hat{\infty} \text{ or vise verse} \\ 0 & \text{if } x = y = \hat{\infty} \end{cases}$$

and D' defines a metric. Furthermore, we define the following OT problem

$$\mathrm{OT}(\hat{\mu},\hat{\nu}) = \inf_{\gamma \in \Gamma(\hat{\mu},\hat{\nu})} \int (D')^p(x,y) d\gamma(x,y)$$

and let $C(\hat{\gamma};\hat{\mu},\hat{\nu})$ to be the corresponding objective function, i.e.

$$C(\hat{\gamma};\hat{\mu},\hat{\nu}) := \int_{\hat{\Omega}} (D')^2(x,y) d\hat{\gamma}(x,y).$$

For each $\gamma \in \Pi_\leq(\mu,\nu)$, we have

$$
\begin{aligned}
&C(\gamma;\mu,\nu,\lambda)\\
&=\int_{\Omega^2}(D^p(x,y)\wedge 2\lambda)d\gamma+\lambda((\mu-\pi_{1\#}\gamma)(\Omega)+(\nu-\pi_{2\#}\gamma)(\Omega))\\
&=\int_{\Omega^2}(D')^p(x,y)d\gamma+\int_{\Omega\times\{\hat{\infty}\}}\lambda d((\mu-\pi_{1\#}\gamma)\otimes\delta_{\hat{\infty}})+\int_{\{\hat{\infty}\}\times\Omega}\lambda d(\delta_{\hat{\infty}}\otimes(\nu-\pi_{2\#}\gamma))+\int_{\{(\hat{\infty},\hat{\infty})\}}0d\delta_{(\hat{\infty},\hat{\infty})}\\
&=\int_{\Omega^2}(D')^p(x,y)d\tilde{\gamma}+\int_{\Omega\times\{\hat{\infty}\}}(D')^p(x,y)d\tilde{\gamma}+\int_{\{\hat{\infty}\}\times\Omega}(D')^p(x,y)d\tilde{\gamma}+\int_{\{(\hat{\infty},\hat{\infty})\}}(D')^p(x,y)d\tilde{\gamma}\\
&=\int_{\tilde{\Omega}^2}(D')^2(x,y)d\tilde{\gamma}\\
&=C(\hat{\gamma};\tilde{\mu},\tilde{\nu}).
\end{aligned}
$$

Combining with the fact $\gamma\mapsto\hat{\gamma}$ is bijection, we have $\mathrm{OPT}_\lambda(\mu,\nu)=\mathrm{OT}(\hat{\mu},\hat{\nu})$.

Choose $\mu_1,\mu_2,\mu_3\in\mathcal{M}_+(\Omega)$ and let $K=\mu_1(\Omega)+\mu_2(\Omega)+\mu_3(\Omega)$. Define $\tilde{\mu}_1,\tilde{\mu}_2,\tilde{\mu}_3$ introduced in section 2 of supplementary material. Since $OT(\cdot,\cdot)^{1/p}$ defines a metric, we have

$$(OT(\tilde{\mu}_1,\tilde{\mu}_3))^{1/p}\le(OT(\tilde{\mu}_1,\tilde{\mu}_2))^{1/p}+(OT(\tilde{\mu}_2,\tilde{\mu}_3))^{1/p}.$$

Therefore:

$$(OPT_\lambda(\mu_1,\mu_3))^{1/p}\le(OPT_\lambda(\mu_1,\mu_2))^{1/p}+(OPT_\lambda(\mu_2,\mu_3))^{1/p}.$$

□

7 Proof of Theorem 5.2

First we claim $\mathrm{SOPT}_\lambda(\cdot,\cdot):(\mathcal{M}_+(\Omega))^2\to\mathbb{R}_+$ is a well defined function. It is clear $\mathrm{SOPT}_\lambda(\cdot,\cdot)$ is a function with domain $\mathcal{M}_+(\Omega)^2$ and co-domain $\mathbb{R}\cup\{\pm\infty\}$. Pick μ,ν, we will show $\mathrm{SOPT}_\lambda\in[0,\infty)$. We have

$$\mathrm{SOPT}_\lambda(\mu,\nu)=\int_{\mathbb{S}^{d-1}}\mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu,(\langle\theta,\cdot\rangle_\#\nu)\,\mathrm{d}\sigma(\theta)\ge 0 \tag{15}$$

where the inequality follows from the fact $\mathrm{OPT}\lambda(\theta)(\langle\theta,\cdot\rangle_\#\mu,(\langle\theta,\rangle_\#\nu)\ge 0,\forall\theta$. It remains to show $\mathrm{SOPT}_\lambda(\mu,\nu)<\infty$. We have

$$
\begin{aligned}
\mathrm{SOPT}_\lambda(\mu,\nu)&\le\int_{\mathbb{S}^{d-1}}\lambda(\theta)(\|\langle\theta,\cdot\rangle_\#\mu\|_{\mathrm{TV}}+\|\langle\theta,\cdot\rangle_\#\nu\|_{\mathrm{TV}})\,\mathrm{d}\sigma(\theta)\\
&=(\mu(\Omega)+\nu(\Omega))\int_{\mathbb{S}^{d-1}}\lambda(\theta)\,\mathrm{d}\sigma(\theta)\\
&<\infty
\end{aligned}
$$

where the first inequality follows by plugging zero measure into the cost function in (3); the second inequality holds since λ is an L^1 function.

Next, we will show $\mu=\nu$ iff $\mathrm{SOPT}_\lambda(\mu,\nu)=0$. If $\mu=\nu$, we have for every θ, $\langle\theta,\rangle_\#\mu=\langle\theta,\rangle_\#\nu$ and thus $\mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu,\langle\theta,\cdot\rangle_\#\nu)=0$. Therefore $\mathrm{SOPT}_\lambda(\mu,\nu)=0$. For the reverse direction, we suppose $\mathrm{SOPT}_\lambda(\mu,\nu)=0$. Since $\mathrm{supp}(\sigma)=\mathbb{S}^{d-1}$, we have for almost every θ, $\mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu,\langle\theta,\cdot\rangle_\#\nu)=0$. For every θ, since $\lambda(\theta)>0$, and $\mathrm{OPT}_{\lambda(\theta)}(\cdot,\cdot)$ is a metric when $p=1$ (see [7, Proposition 2.10] or [16, Proposition 5]) or the p-th power of a metric (see theorem 6.1, or [13, Theorem 2.2])when $p\ge 1$, we have $\langle\theta,\cdot\rangle_\#\mu=\langle\theta,\cdot\rangle_\#\nu$. By the injectivity of Radon transform on measures (see [2, Proposition 7]), we have $\mu=\nu$.

For symmetry, we have

$$
\begin{aligned}
\mathrm{SOPT}_\lambda(\mu,\nu)&=\int_{\mathbb{S}^{d-1}}\mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu,\langle\theta,\rangle_\#\nu)\,\mathrm{d}\sigma(\theta)\\
&=\int_{\mathbb{S}^{d-1}}\mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\nu,\langle\theta,\cdot\rangle_\#\mu)\,\mathrm{d}\sigma(\theta)\\
&=\mathrm{SOPT}_\lambda(\nu,\mu)
\end{aligned}
\tag{16}
$$

where the second equality follows from the fact $\mathrm{OPT}_{\lambda(\theta)}(\cdot,\cdot)$ is a metric for each θ.

For the triangle inequality, we choose μ_1, μ_2, μ_3 and have:

$$
\begin{aligned}
&\mathrm{SOPT}_\lambda(\mu_1,\mu_3)^{1/p} \\
&= \left(\int_{\mathbb{S}^{d-1}} \mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu_1, \langle\theta,\cdot\rangle_\#\mu_3)\,\mathrm{d}\sigma(\theta)\right)^{1/p} \\
&\le \left\{\int_{\mathbb{S}^{d-1}} \left[(\mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu_1, \langle\theta,\cdot\rangle_\#\mu_2)^{1/p} + (\mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu_2, \langle\theta,\cdot\rangle_\#\mu_3))^{1/p}\right]^p \mathrm{d}\sigma(\theta)\right\}^{1/p} \\
&\le \left(\int_{\mathbb{S}^{d-1}} \mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu_1, \langle\theta,\cdot\rangle_\#\mu_2)\,\mathrm{d}\sigma(\theta)\right)^{1/p} + \left(\int_{\mathbb{S}^{d-1}} \mathrm{OPT}_{\lambda(\theta)}(\langle\theta,\cdot\rangle_\#\mu_1, \langle\theta,\cdot\rangle_\#\mu_2)\,\mathrm{d}\sigma(\theta)\right)^{1/p} \\
&= \mathrm{SOPT}_\lambda(\mu_1,\mu_2)^{1/p} + \mathrm{SOPT}_\lambda(\mu_2,\mu_3)^{1/p}
\end{aligned}
$$

where the first inequality follows from the fact $\mathrm{OPT}_{\lambda(\theta)}(\cdot,\cdot)$ is the p-th power of a metric for each θ; the second inequality follows from Minkowski inequality in $L_p(\mathbb{S}^{d-1})$.

8 Application in Color Adaptation

Transferring colors between images is a classical task in computer vision and image science. Given two images, the goal is to impose on one of the images (source image) the histogram of the other image (target image). Optimal transport-based approaches have been developed and achieved great success in this task [6, 2, 18, 9]. However, in the balanced OT setting, the OT-based approach requires normalizing the histograms of (sampled) colors, which may lead to undesired performance. For example, suppose the majority of a target image is red, (e.g. an image of evening sunset) and the majority of a source image is green (e.g. image of trees). Then balanced-OT-based approaches will produce a red tree in the result. To address this issue, [1] applied the SPOT-based approach which will match all the pixels in the source image to *partial* pixels in the target image.

Our method. Inspired by [9, 1], our method contains the following three steps: First, we sample pixels from the source and target image by k-mean clustering (or another sampling method). Second, we transport the sampled source pixels into the target domain. If OT or entropic OT is applied, it can be done by the optimal transportation plan; if sliced-OT is applied, the source pixels would be updated iteratively for each slice based on the gradient of 1-D OT with respect to the source pixels (see equation (46) in [2], or line 5 in our algorithm 3). In our method, we apply the transportation plan from OPT. Third, we reconstruct the source image based on the transported source pixels (e.g. see Equation 4.1 in [9]).

Experiment. We first normalize all the pixels in the source and target images to be in range $[0, 1]$, then we use k-means clustering to sample 5000 pixels from the source image and 10000 pixels from the target image. We compare the performance of the OT-based and Entropic-OT based domain adaptation functions in PythonOT [12] (ot.da.EMDTransport and ot.da.SinkhornTransport) whose OT's solver is written in C++ [1], SPOT [1] and our method based on sliced optimal partial transport. For our method, we test it in two schemes, $\lambda = 10.0$ and $\lambda < 2.0$. In the first case, λ achieves the maximum of distance of two (normalized) pixels, that is, we will transport all the source pixels into target domain. In second case, we choose λ and theoretically, only partial source pixels will be transported into the target domain.

Performance. In these examples, the OT-based approach which matches all (sampled) pixels of the source image to all pixels of a target image can lead to undesired results. For example, in the second row of Figure 1, the third image has dark blue on the sky and red color on the ground. This issue is alleviated in SPOT and our method. In our method, when $\lambda = 5.0$, we will transfer all the (sampled) pixels from source to target and the result is similar to the result of SPOT [2]. When $\lambda < 2.0$, the result image is closer to the source image. OT-based method requires 40-50 seconds (we set the maximum iteration number of linear programming to be 1000,000); Partial OT method requires 80-90 seconds (the # of projections is set to be 400) and our method requires 60-80 seconds (the # of projections is set to be 400). The data type is 32-bit float number and the experiment is conducted on a Linux computer with AMD EPYC 7702P CPU with 64 cores and 256GB DDR4 RAM.

[1] We modify their code to increase the speed.

[2] We conjecture the two results are not exactly the same due to the randomness of projections.

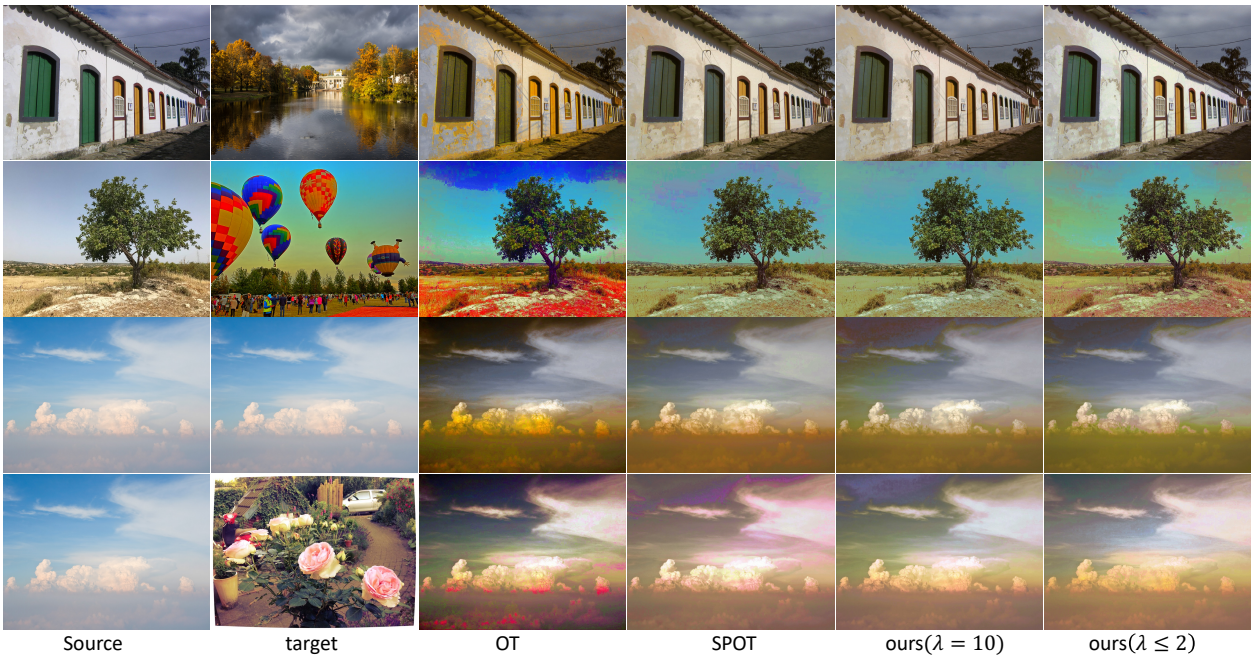

Figure 1: We transfer colors from source image to the target image by the methods based on optimal transport [9], SPOT [1] and our SOPT. For our method, we set $\lambda = 10$ and a small value less than 2. Image via Flickr: Facade by Phil Whitehouse, palace by Neil Williamson, clouds by Tim Wang, air balloon by Kirt Edblom, roses by Felix Schaumburg.

References

[1] Nicolas Bonneel and David Coeurjolly. SPOT: sliced partial optimal transport. *ACM Transactions on Graphics*, 38(4):1–13, 2019. 11, 12, 14

[2] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. 10, 11

[3] R. E. Burkhard, B. Klinz, and R. Rudolf. Perspectives of Monge properties in optimization. *Discr. Appl. Math.*, 70(2):95–161, 1996. 4

[4] Luis A Caffarelli and Robert J McCann. Free boundaries in optimal transport and monge-ampere obstacle problems. *Annals of mathematics*, pages 673–730, 2010. 1

[5] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, 18(1):1–44, 2018. 1

[6] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018. 11

[7] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and Kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018. 1, 10

[8] Shaoyi Du, Nanning Zheng, Shihui Ying, Qubo You, and Yang Wu. An extension of the icp algorithm considering scale factor. In *2007 IEEE International Conference on Image Processing*, volume 5, pages V–193. IEEE, 2007. 14

[9] Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-François Aujol. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014. 11, 12

[10] Alessio Figalli. The optimal partial transport problem. *Archive for rational mechanics and analysis*, 195(2):533–560, 2010. 1

[11] Alessio Figalli and Nicola Gigli. A new transportation distance between non-negative measures, with applications to gradients flows with dirichlet boundary conditions. *Journal de mathématiques pures et appliquées*, 94(2):107–130, 2010. 1

[12] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud,

Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 11

[13] Florian Heinemann, Marcel Klatt, and Axel Munk. Kantorovich–rubinstein distance and barycenter for finitely supported measures: Foundations and algorithms. *Applied Mathematics & Optimization*, 87(1):4, 2023. 9, 10

[14] Wonjun Lee, Rongjie Lai, Wuchen Li, and Stanley Osher. Generalized unnormalized optimal transport and its fast algorithms. *Journal of Computational Physics*, 436:110041, 2021. 1

[15] Matthias Liero, Alexander Mielke, and Giuseppe Savare. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018. 1, 3, 4

[16] Benedetto Piccoli and Francesco Rossi. Generalized wasserstein distance and its application to transport equations with source. *Archive for Rational Mechanics and Analysis*, 211(1):335–358, 2014. 1, 10

[17] Benedetto Piccoli and Francesco Rossi. On properties of the generalized wasserstein distance. *Archive for Rational Mechanics and Analysis*, 222(3):1339–1365, 2016. 1

[18] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011. 11

[19] Filippo Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser Boston, 2015. 3

[20] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 14

[21] Cedric Villani. *Optimal transport: old and new*. Springer, 2009. 2, 3

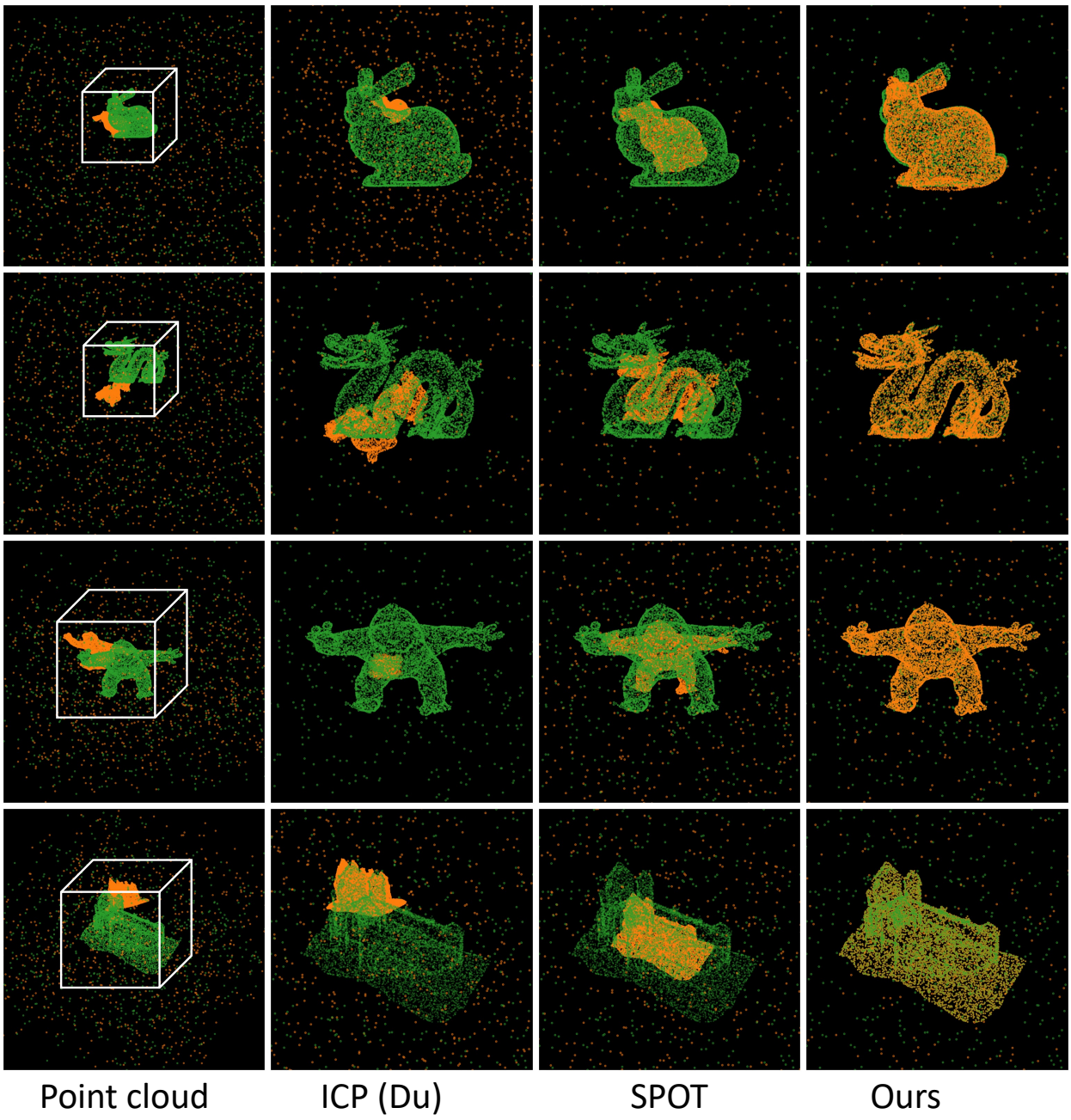

Figure 2: We visualize the results of ICP (Du) [8], ICP (Umeyama) [20], SPOT [1] and our method. The datasets contains Stanford Bunny, Stanford dragon, Mubble sitting and Witch Castle.